Analysis of Waiting-Time Predictability in Scalable Media Streaming

Mohammad Alsmirat, Musab Al-Hadrusi, and Nabil J. Sarhan.

ACM Multimedia 2007 Conference Presentation

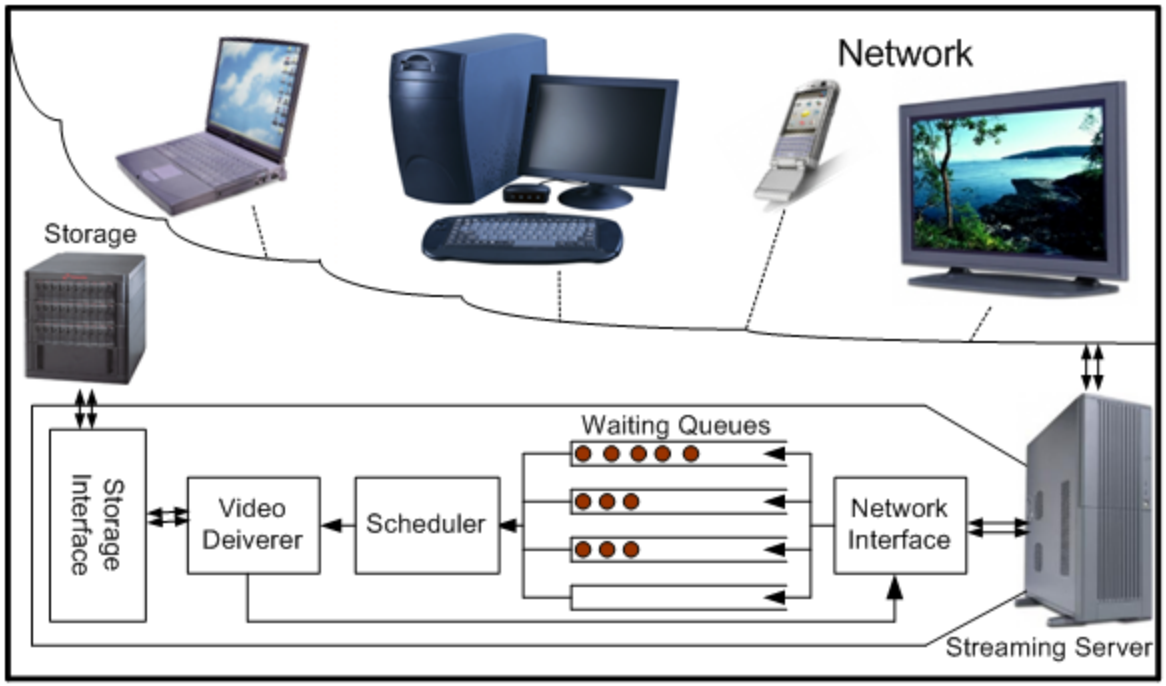
This work has been supported in part by NSF grant CNS-0626861.
This paper is published as [25].
A journal paper of this conference paper was published at ACM TOMCCAP 2010 [26].

Outline

- Introduction
- Related Work
- Main Contributions
- Waiting Time Prediction
- Performance Evaluation and Main Results
- Conclusions

Introduction

- The distribution of streaming media faces a significant scalability challenge [17, 18. 19, 20]
- This challenge has been addressed by Media Delivery *and* Request Scheduling.



Introduction (Cont…)

- Prior work focused on server throughput, average waiting time, and unfairness.
- We consider another metric: ability to inform users with expected waiting times. Why?
  - Growing interest in human-centered multimedia.
  - Online video users may experience significant delays.
  - Larger delay for long high-quality videos
  - Feedback encourages waiting.
- We propose waiting-time prediction.

Related Work

- Video Delivery Strategies:
  - Periodic Broadcasting [12,14,2,13,23]
  - **Stream Merging** [6,8,10,4,16]
    - Patching
    - Transition Patching
    - ERMT

Related Work (Cont…)

- Request Scheduling
  - One waiting queue for each video
  - Main Scheduling Policies:

Policy	Selects the queue with
First Come First Serve (FCFS) [7]	oldest request
Maximum Queue Length (MQL) [7]	largest number of requests
Maximum Factored Queue Length (MFQL) [1]	largest factored length
Minimum Cost First [22]	least cost (per request)
Next Schedule Time First (NSTF) [21]	earliest schedule time

Main Contributions

- Extending NSTF to work with stream merging.
- Proposing the waiting-time prediction approach.
- Presenting three prediction schemes.
- Conducing extensive analysis
  - Patching, Transition Patching, ERMT
  - FCFS, MQL, MCF-P (RAP), MCF-P (RAF)
  - Several performance metrics
  - Several workload and design parameters

Waiting-Time Prediction

- The proposed waiting-time prediction approach provides users with expected waiting times for service rather than hard time-of-service guarantees.
- It overcomes the shortcomings of NSTF.
  - NSTF may not perform well.
  - NSTF cannot work with hierarchical stream merging.
- The server can use highly scalable scheduling policies, such as MCF, while improving QoS.

Waiting Time Prediction (Cont…)

- It can be applied with any stream merging technique and any scheduling policy.
- We propose three schemes for predicting waiting time
  - *Assign Overall Average Waiting Time (AOW)*
  - *Assign Per-Video Average Waiting Time (APW)*
  - *Assign Expected Stream Completion Time (AEC)*

Proposed AEC Scheme

- Predicts the waiting times by “simulating” the future behavior of the server.
- Utilizes detailed information about the current server state and considers the scheduling policy.
- This Information includes
  - current queue lengths
  - completion times of running streams
  - average request arrival rate for each video

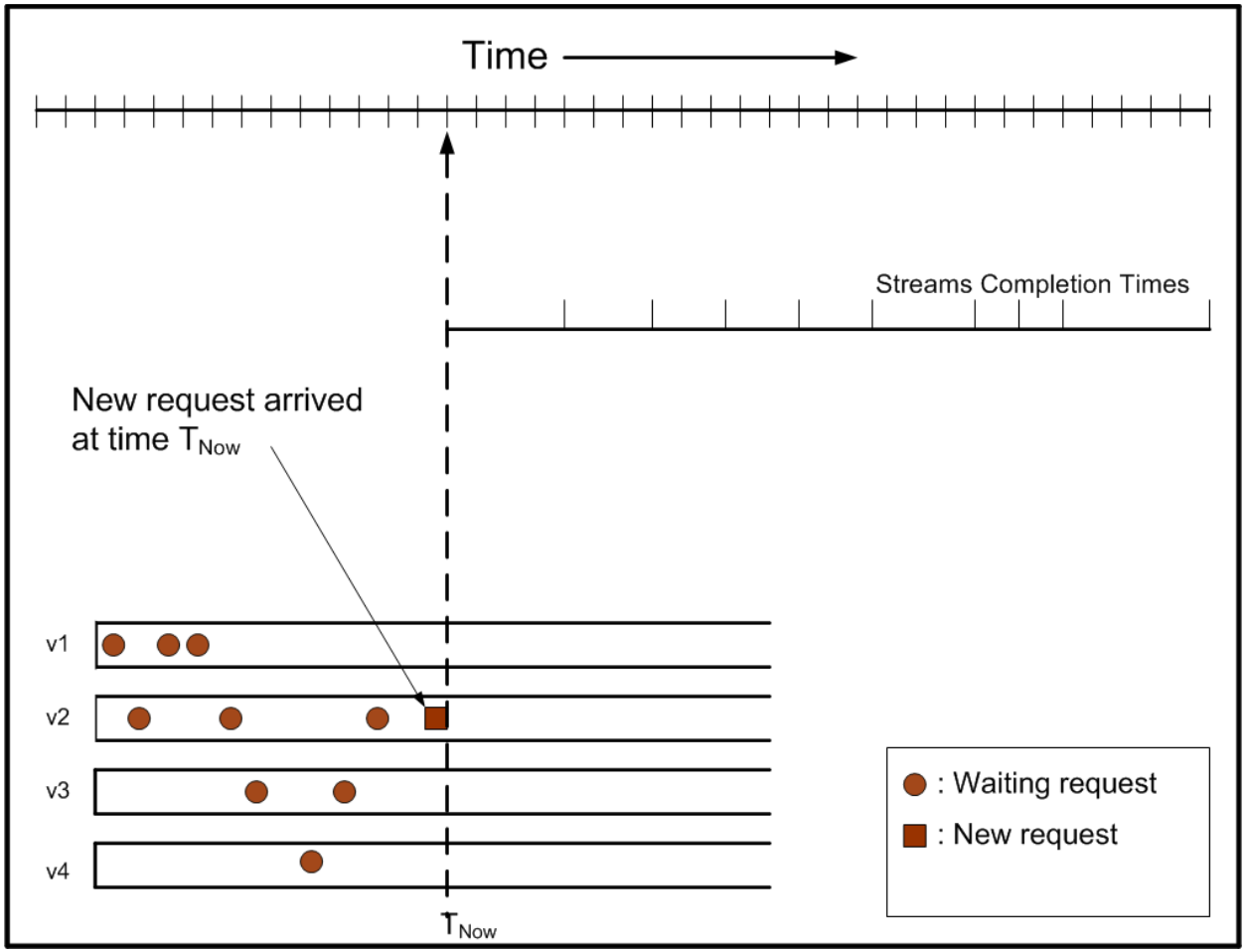
Proposed AEC Scheme (Cont…)

- To reduce the implementation complexity, AEC predicts the future scheduling decisions during only certain duration of time, called *prediction window (Wp).*

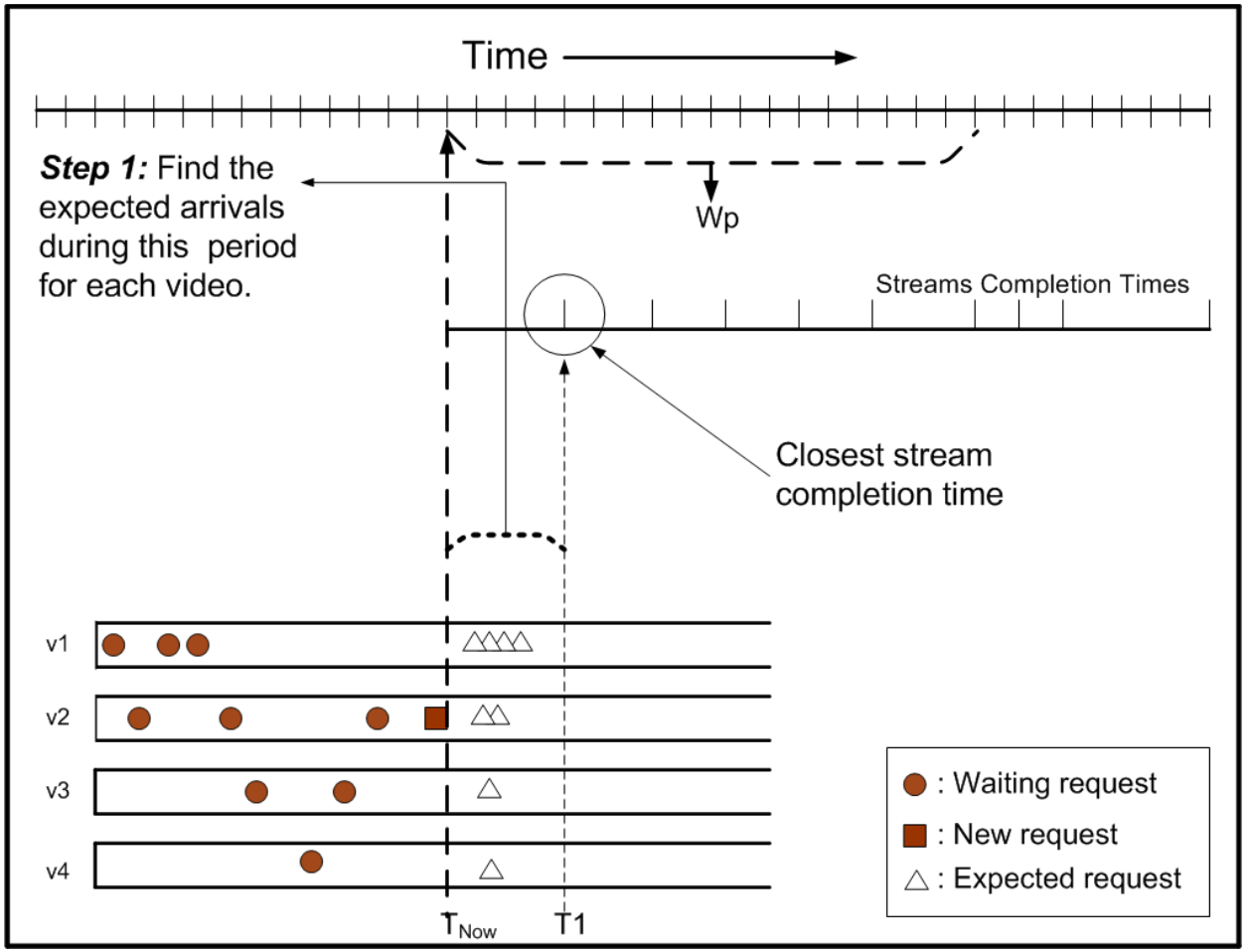
Simplified AEC Algorithm

```
for (v = 0; v < Nv ; v ++) // Initialize the assigned time for each video
    assigned_time[v] = −1;
T = closest completion time; // Start with closest completion time
while (T < TNow + Wp) { // Loop till prediction window is exceeded
    // Find expected video queue lengths
    for (v = 0; v < Nv ; v + +){
        if (video v has not been assigned an expected time)
            expected_qlen[v] = qlen[v] + [v] × (T − TNow);
        else
            expected_qlen[v] = [v] × (T − assigned_time[v]);
        Compute scheduling objective function for video v;
    } //for
    // Find the expected video to be served at time T
    expected_video = find video with minimum objective function;
    if (expected_video == v j ){
        Assign T to request Ri as the expected service time
        break; // Done
    } else
        assigned_time[expected_video] = T ;
    T = next completion time; // Try again for the completion time
} //while
```

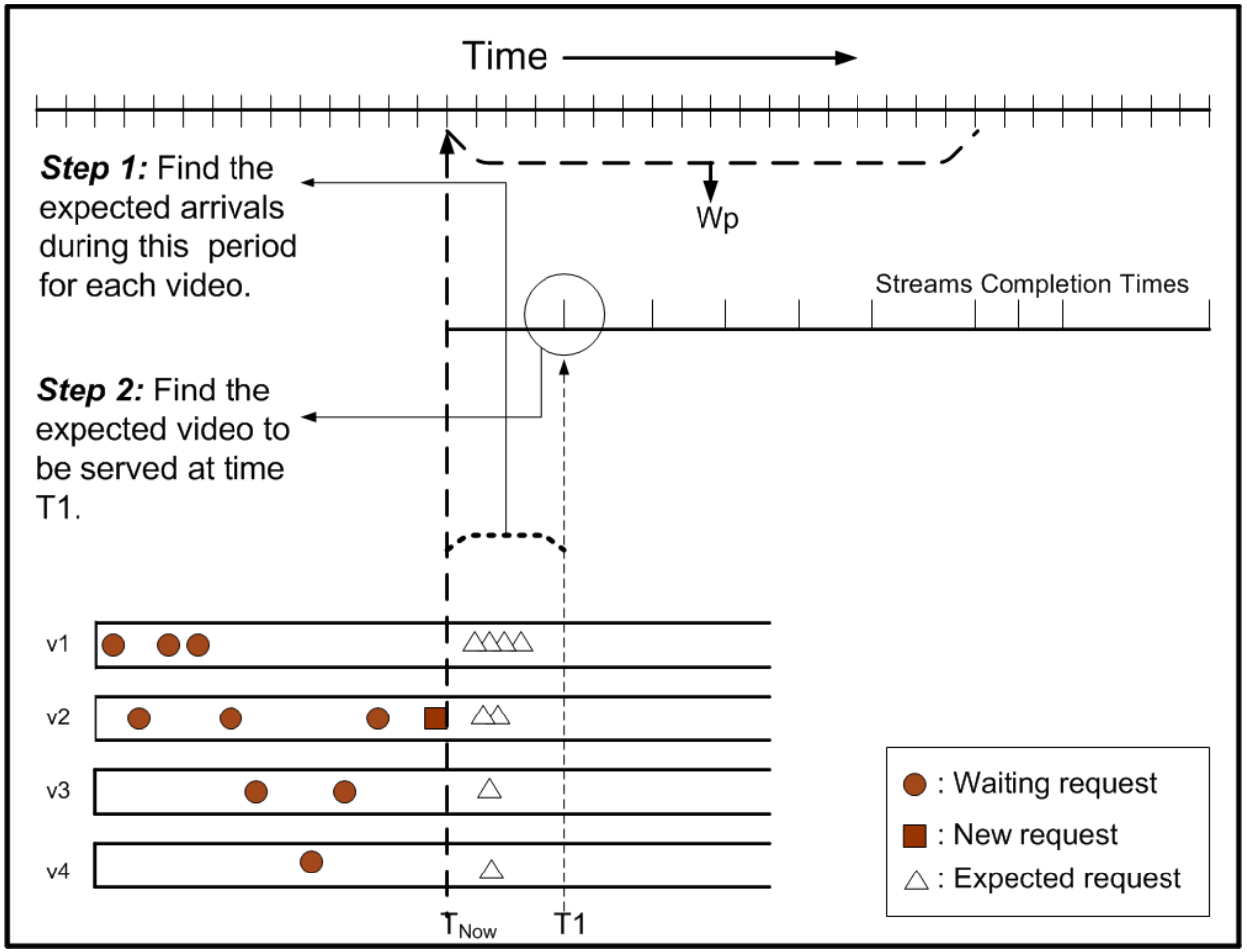
AEC Clarification Example



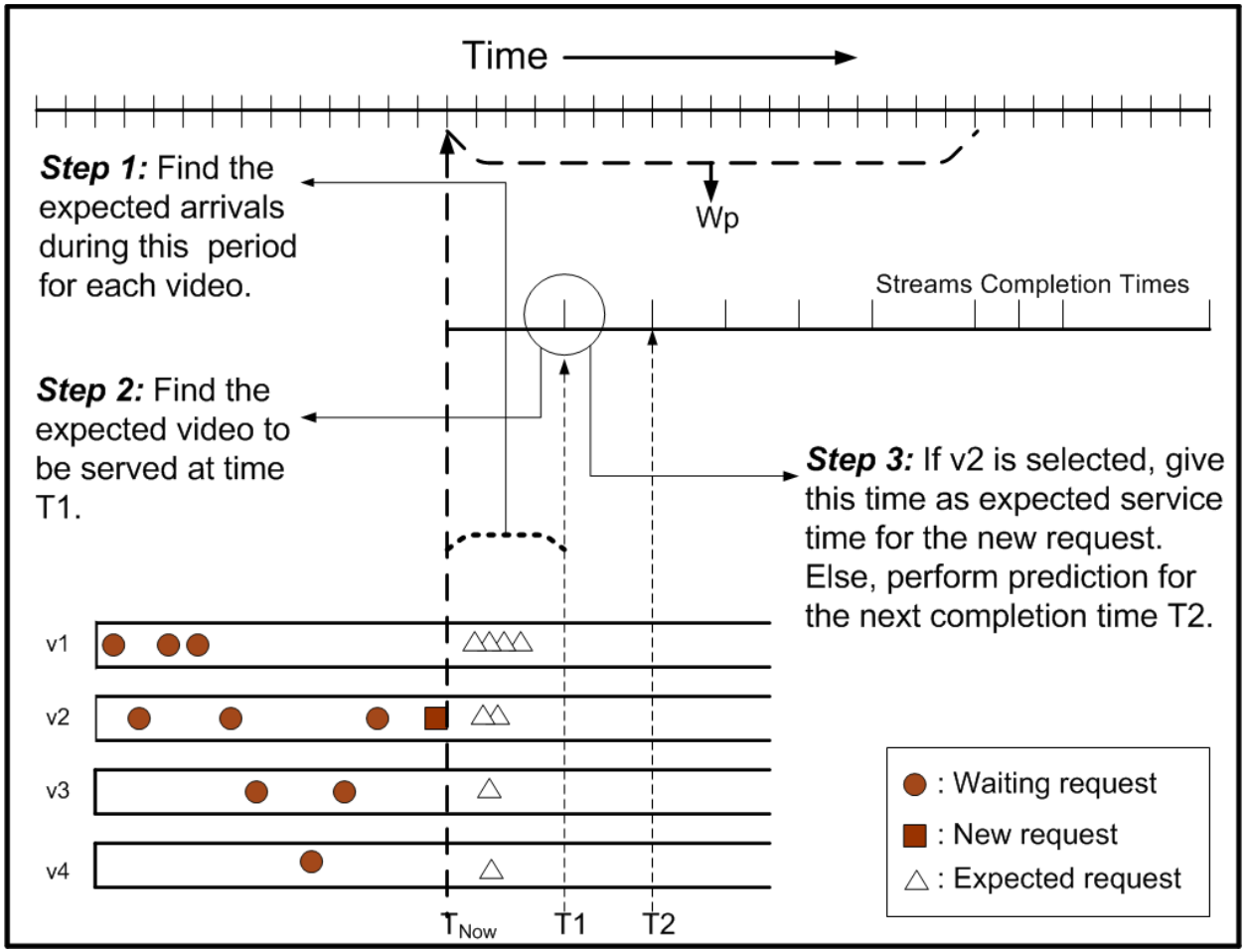
AEC Clarification Example (Cont…)



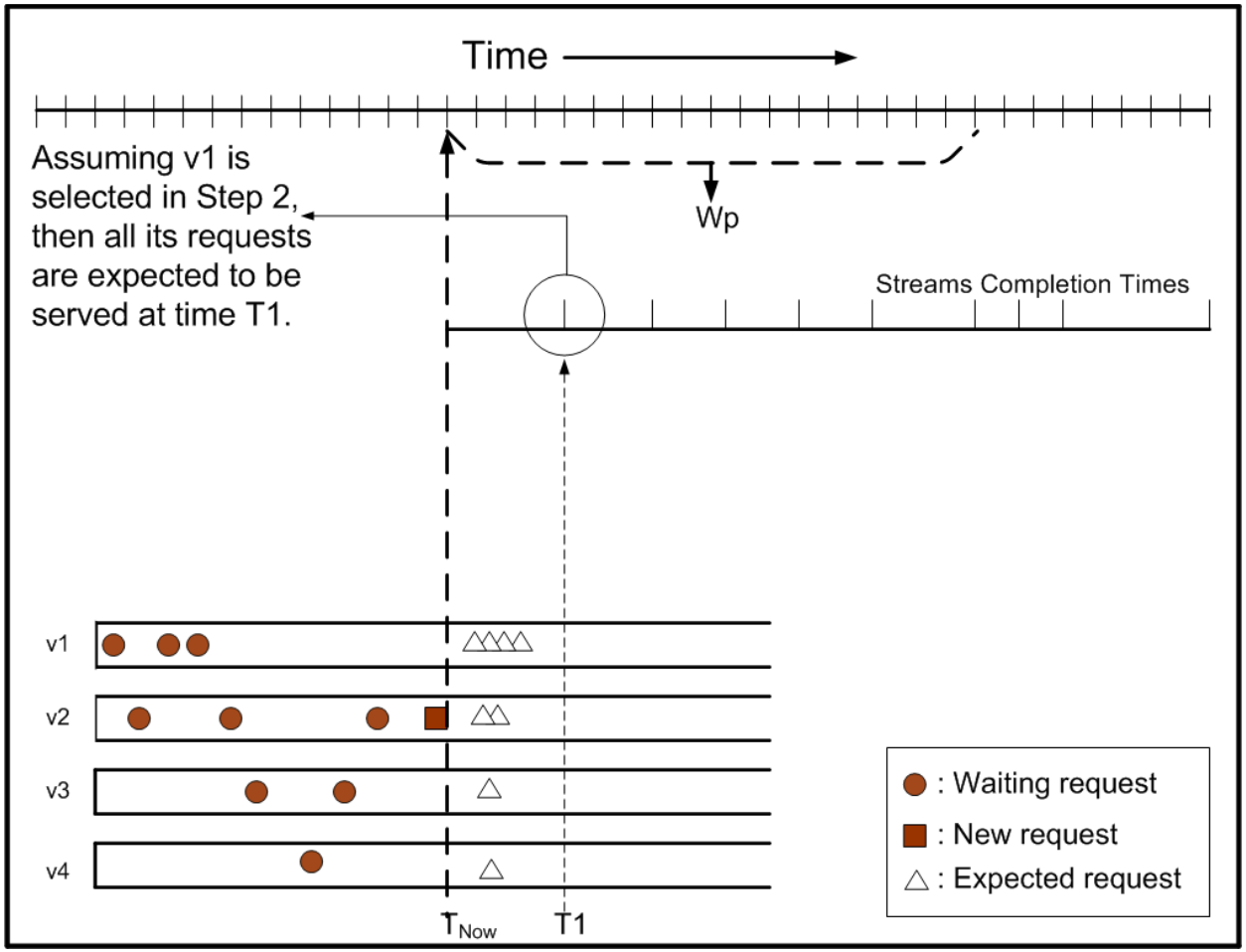
AEC Clarification Example (Cont…)



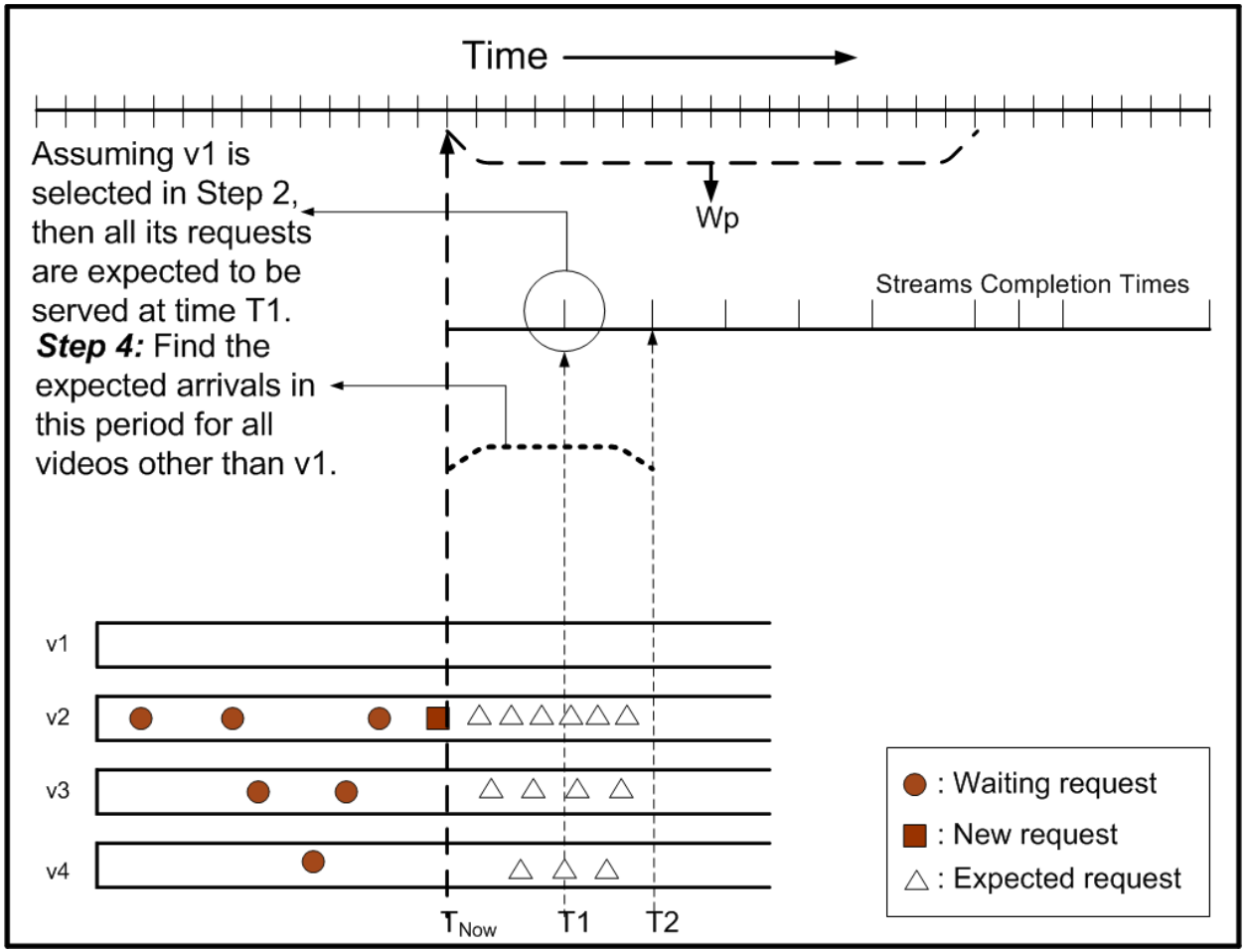
AEC Clarification Example (Cont…)



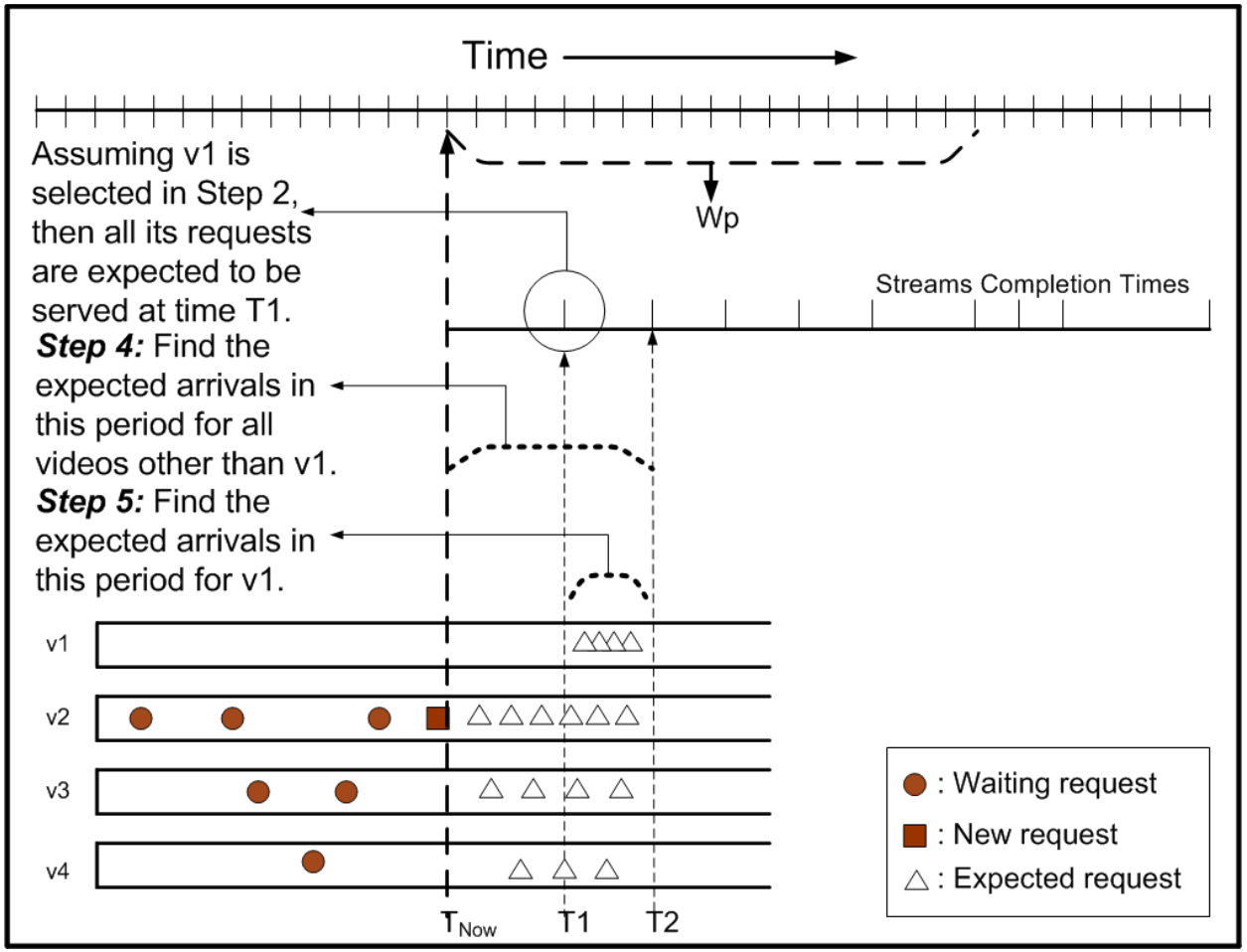
AEC Clarification Example (Cont…)



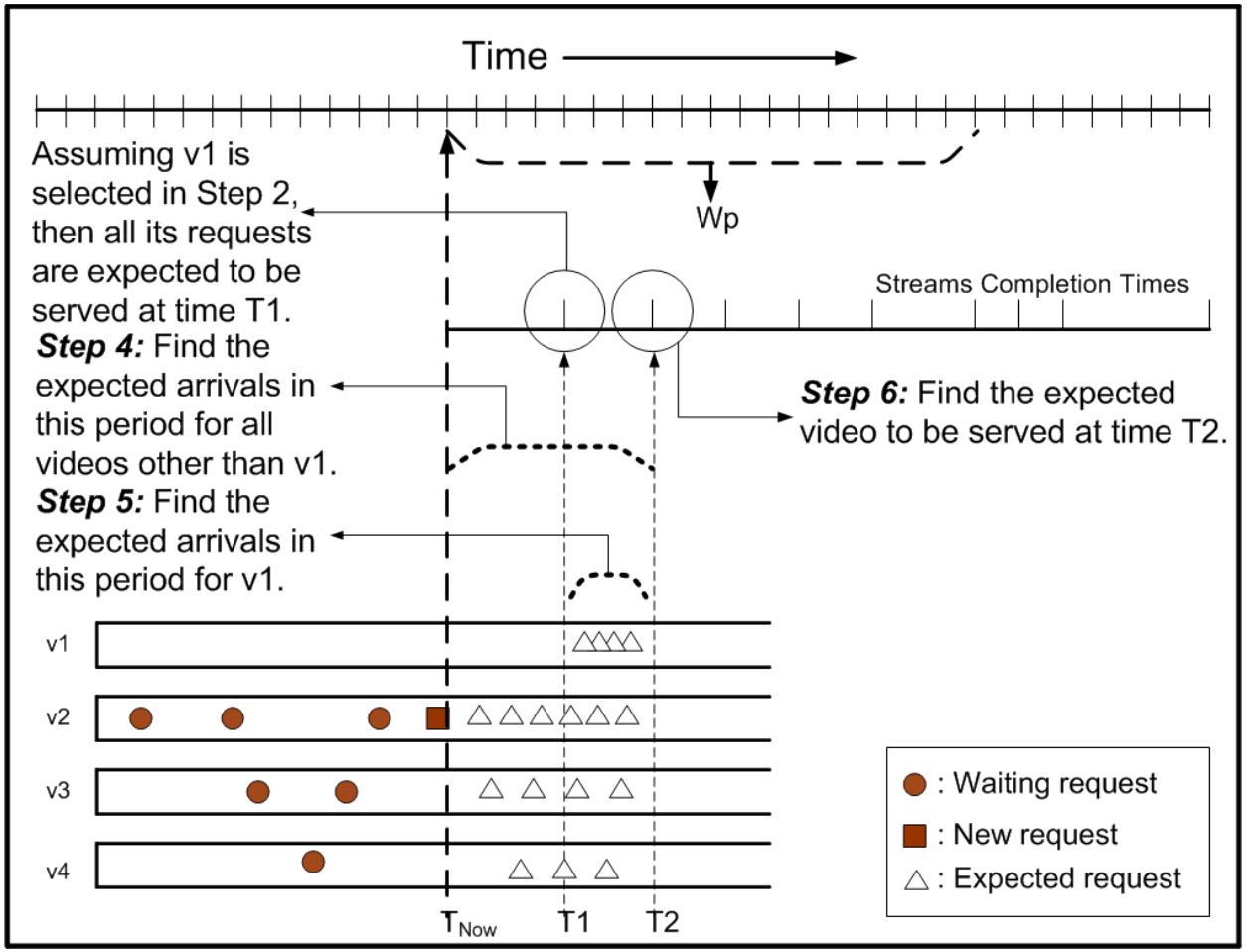
AEC Clarification Example (Cont…)



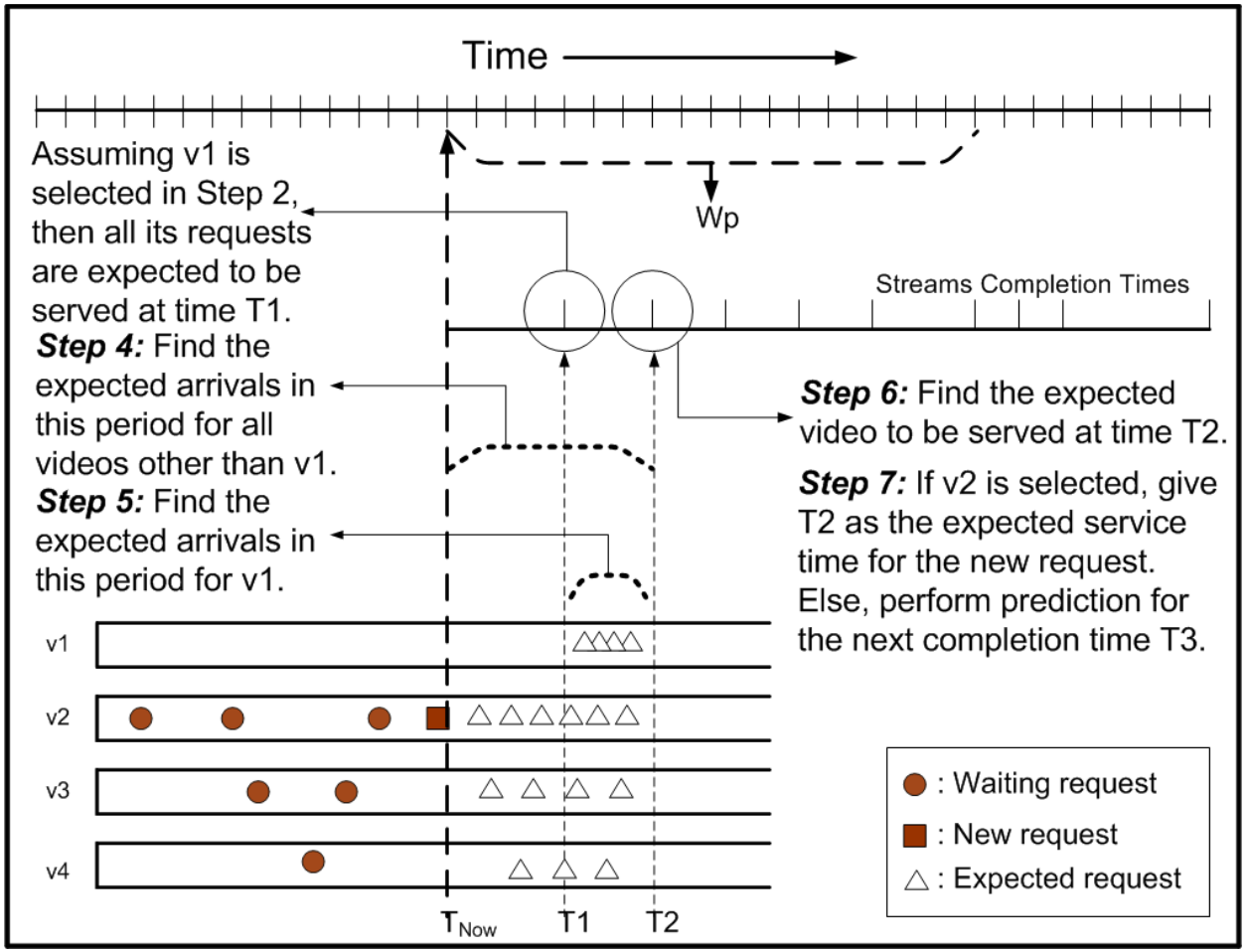
AEC Clarification Example (Cont…)



AEC Clarification Example (Cont…)



AEC Clarification Example (Cont…)



Enhancements of AEC

- *Enhancement 1: Predict Stream Completion Times*
- *Enhancement 2: Account for User Defections*
- *Enhancement 3: Give Preference to Real Requests*

Performance Evaluation and Main Results

Workload Characteristics

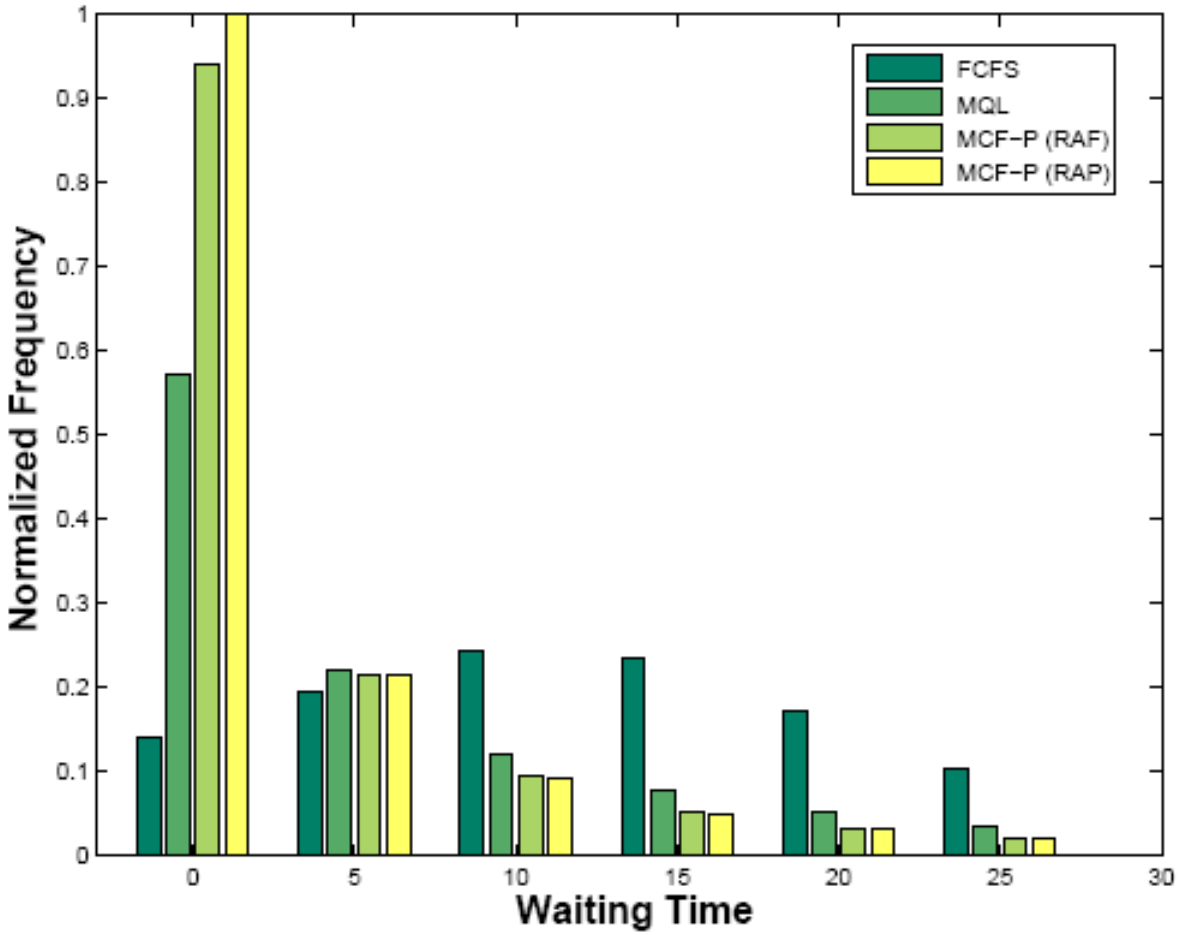
Parameter	**Model/Value(s)**
Request Arrival	Poisson Process
Request Arrival Rate	Variable, Default = 40 Req./min
Server Capacity	150 to 850 channels
Video Access	Zipf-Like Skewness Parameter $\theta = 0.271$
Number of Videos	Variable, Default = 120
Video Length	Variable, Default 120 min
Waiting Tolerance Model	Models A, B, and C
Mean Waiting Tolerance (μ_{tol})	Variable, Default = 30 sec

Performance Metrics

- Prediction accuracy
  - measured as average deviation between the expected and actual times of service
- Percentage of clients receiving expected times of service (PCRE)
- Customer defection (turn-away) probability
- Average request waiting time
- Unfairness

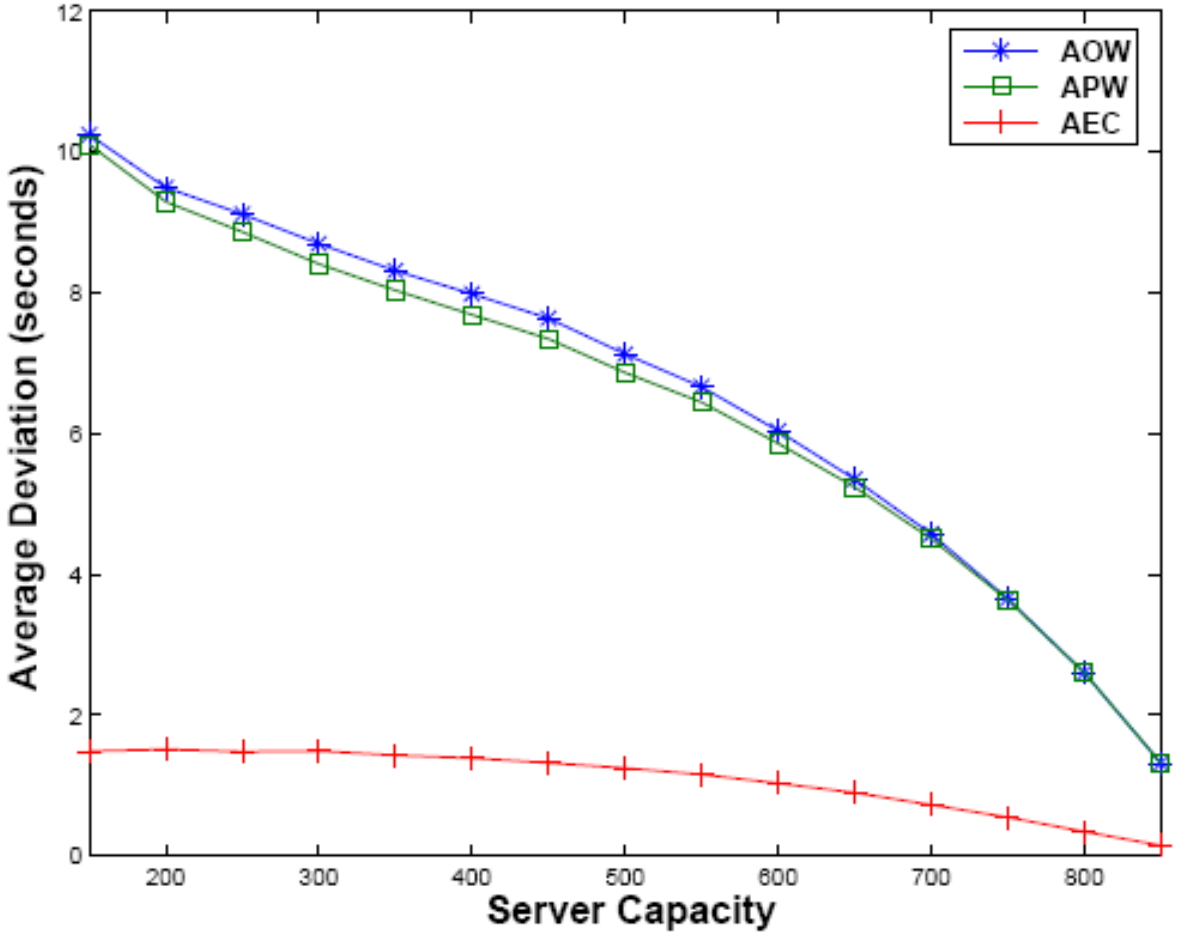
Main Results

Waiting Time Distribution



- *Patching*
- *600 Channels*
- *Model A*
- *All videos*

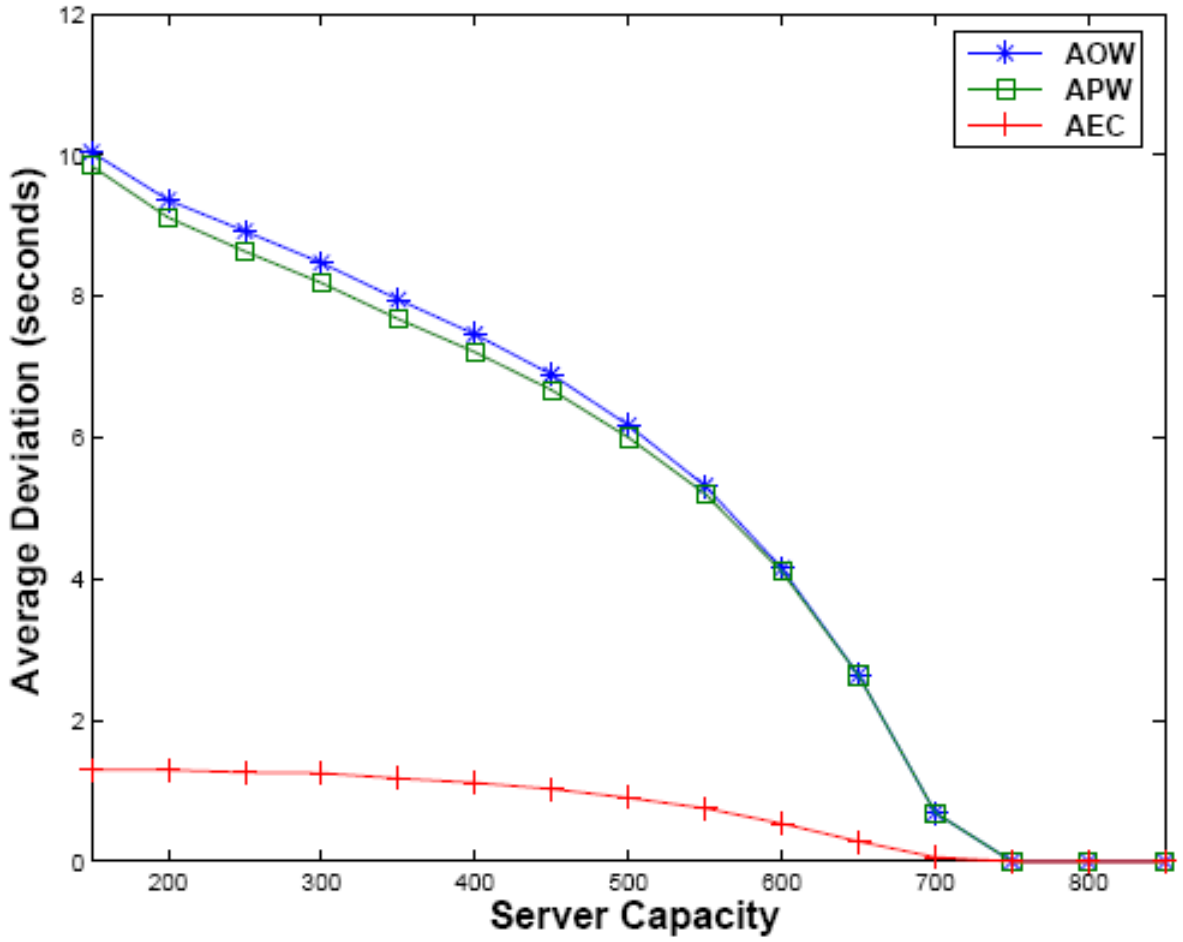
Effectiveness of Prediction Schemes



- *Patching*
- *MCF-P(RAP)*
- *Wp =0.5µtol*
- *Model A*

Effectiveness of Prediction Schemes

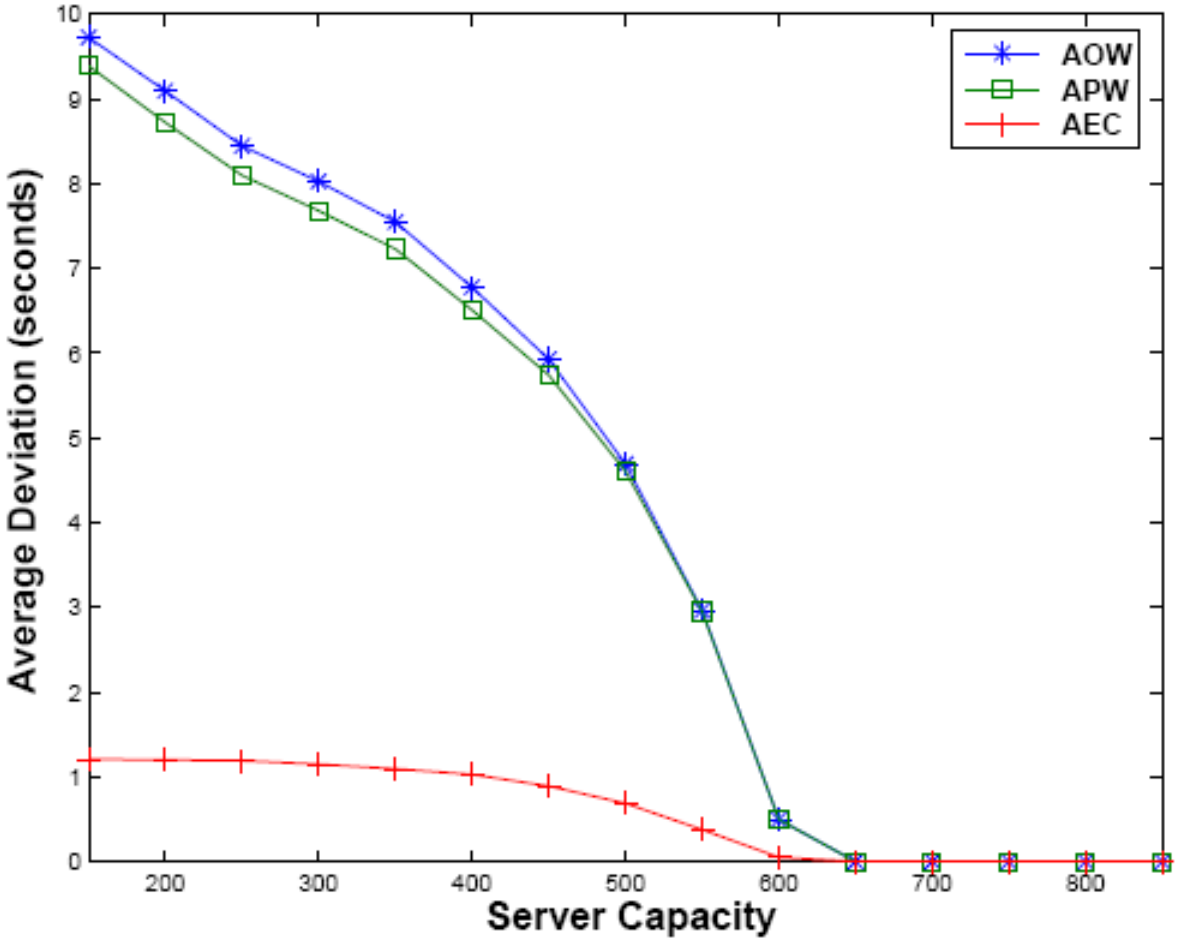
(cont…)



- *Transition Patching*
- *MCF-P(RAP)*
- *Wp =0.5µtol*
- *Model A*

Effectiveness of Prediction Schemes

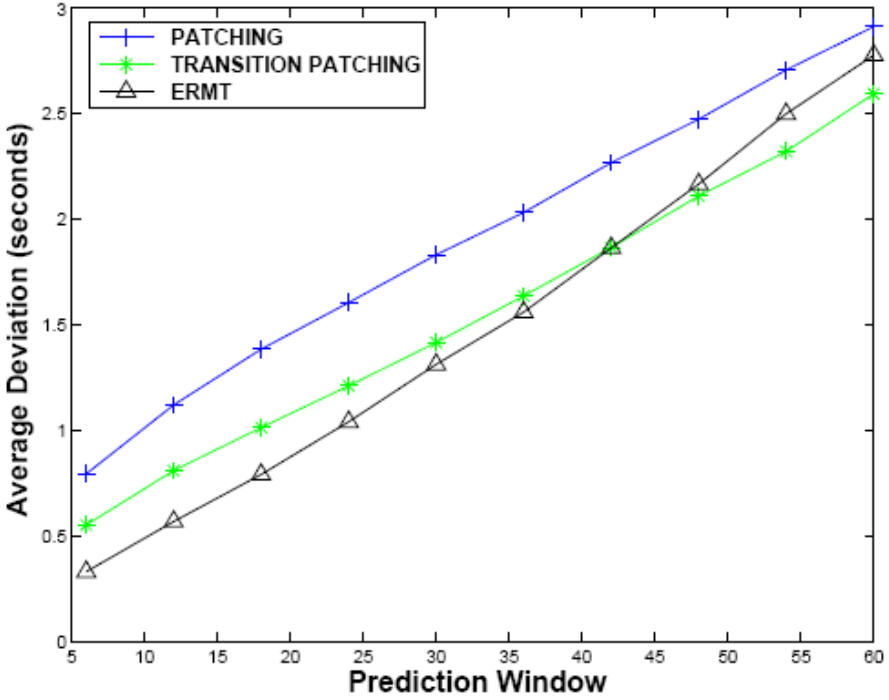
(cont…)

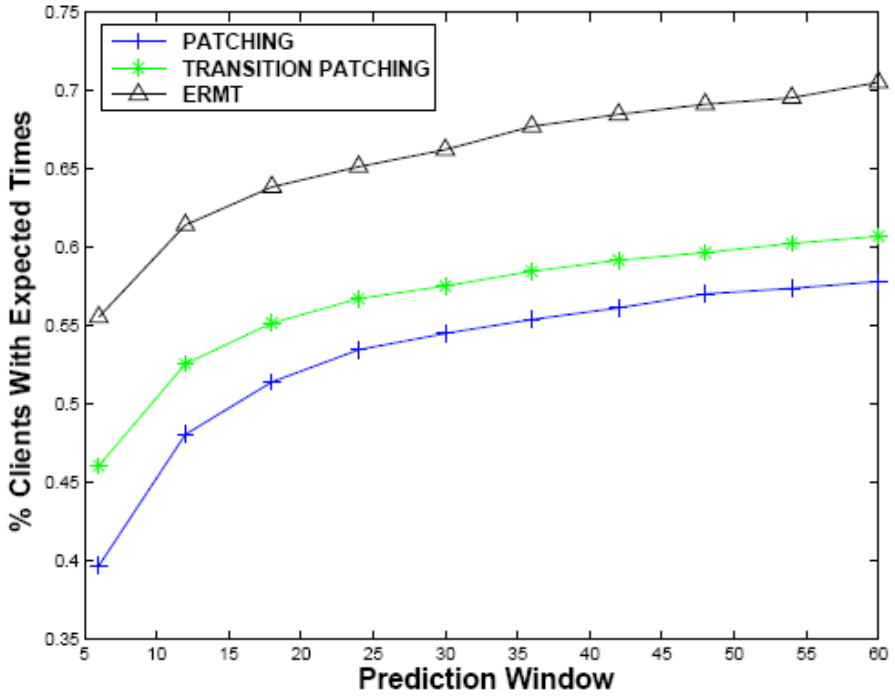


- *ERMT*
- *MCF-P(RAP)*
- *Wp =0.5µtol*
- *Model A*

Impact Of Prediction Window

[AEC, MCF-P(RAP), 500 Channels, Model A]

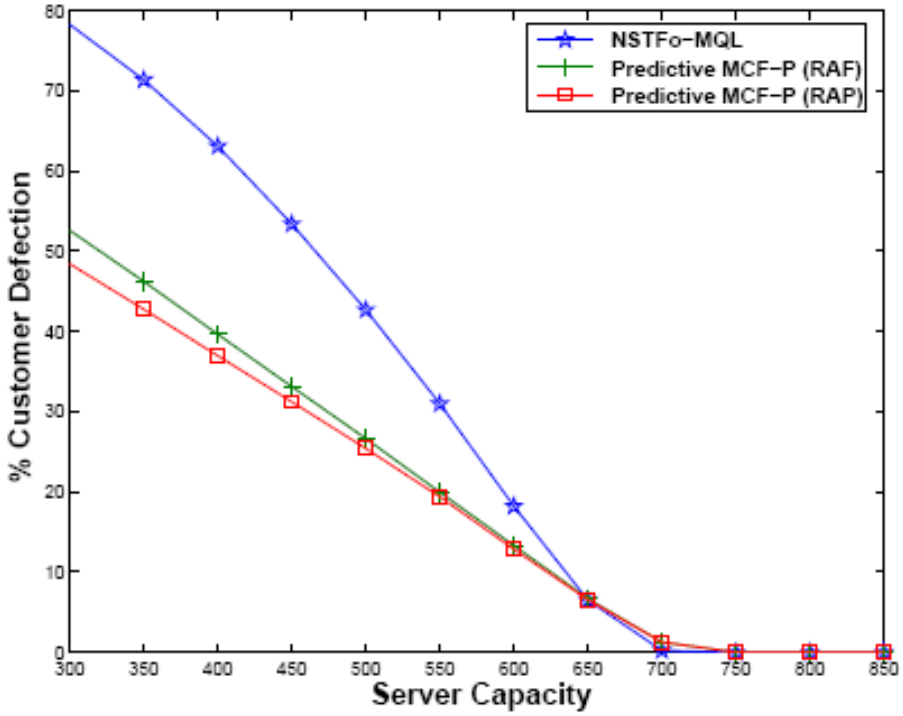


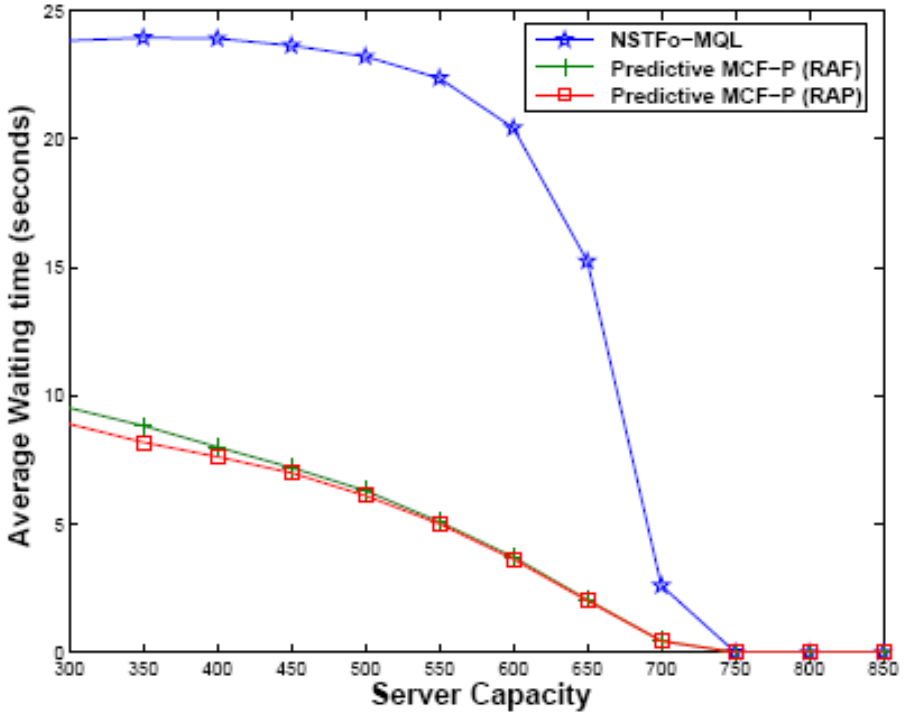


Effectiveness of the Predictive Approach Compared with NSTF (cont…)

Comparing NSTF with Predictive MCF-P (RAP) and MCF-P (RAF)

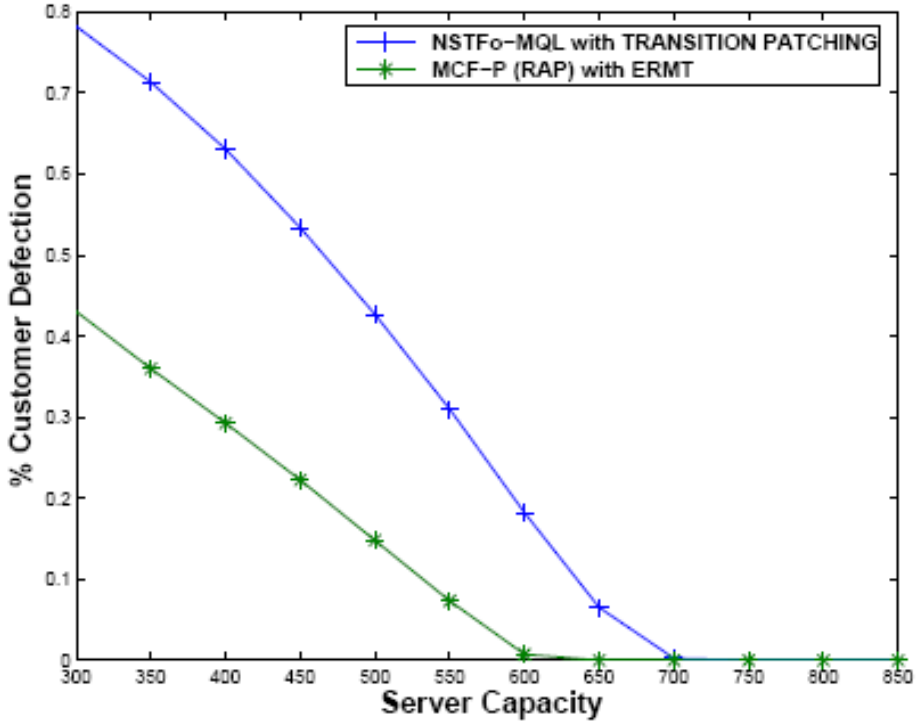
[Transition Patching, Wp = 0.5µtol , Model C]

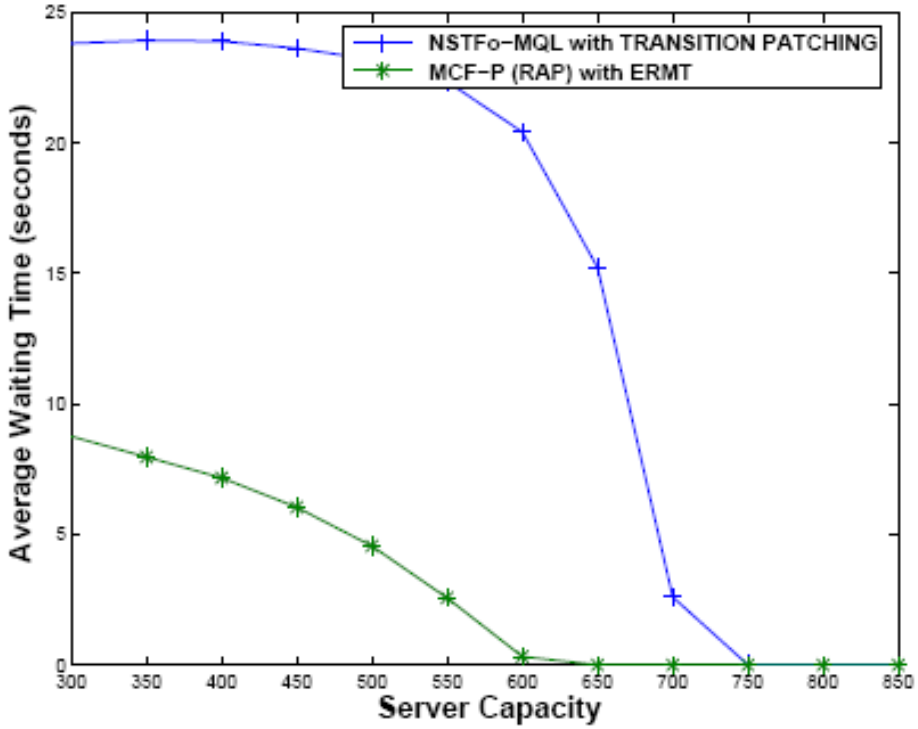




Effectiveness of the Predictive Approach Compared with NSTF (cont…)

Comparing NSTF and Predictive MCF-P, Each with Its Most Scalable Stream Merging Technique [Model C]





Conclusions

- We have proposed a predictive approach of the user waiting time.
- We have presented three prediction schemes: AOW, APW, and AEC.
- The waiting time can be predicted accurately.
- MCF-P is not only highly predictable but also achieves the best performance in server throughput and average waiting time.
- Combining AEC with MCF-P leads to outstanding performance benefits, compared with NSTF.

References

[1] C. C. Aggarwal, J. L. Wolf, and P. S. Yu. The maximum factor queue length batching scheme for Video-on-Demand systems. IEEE Trans. on Computers, 50(2):97–110, Feb. 2001.

[2] A. Bar-Noy, G. Goshi, R. Ladner, and K. Tam. Comparison of stream merging algorithms for Media-on-Demand. Multimedia Systems Journal, 9:211–223, 2004.

[3] Y. Cai and K. A. Hua. An efficient bandwidth-sharing technique for true video on demand systems. In Proc. of ACM Multimedia, pages 211–214, Oct. 1999.

[4] Y. Cai and K. A. Hua. Sharing multicast videos using patching streams. Multimedia Tools and Applications Journal, 21(2):125–146, Nov. 2003.

[5] Y. Cai, W. Tavanapong, and K. A. Hua. Enhancing patching performance through double patching. In Proc. of 9th Int'l
Conf. on Distributed Multimedia Systems, pages 72–77, Sept. 2003.

[6] S. W. Carter and D. D. E. Long. Improving Video-on-Demand server efficiency through stream tapping. In the Int'l Conf. on Computer Communication and Networks (ICCCN), pages 200–207, Sept. 1997.

[7] A. Dan, D. Sitaram, and P. Shahabuddin. Scheduling policies for an on-demand video server with batching. In Proc. Of ACM Multimedia, pages 391–398, Oct. 1994.

References

[8] D. L. Eager, M. K. Vernon, and J. Zahorjan. Optimal and efficient merging schedules for Video-on-Demand servers. In Proc. of ACM Multimedia, pages 199–202, Oct. 1999.

[9] D. L. Eager, M. K. Vernon, and J. Zahorjan. Bandwidth skimming: A technique for cost-effective Video-on-Demand. In Proc. of Multimedia Computing and Networking Conf. (MMCN), pages 206–215, Jan. 2000.

[10] D. L. Eager, M. K. Vernon, and J. Zahorjan. Minimizing bandwidth requirements for on-demand data delivery. IEEE
Trans. on Knowledge and Data Engineering, 13(5):742–757, Sept. 2001.

[11] K. A. Hua, Y. Cai, and S. Sheu. Patching: A multicast technique for true Video-on-Demand services. In Proc. of
ACM Multimedia, pages 191–200, 1998.

[12] K. A. Hua and S. Sheu. Skyscraper broadcasting: A new broadcasting scheme for metropolitan Video-on-Demand
system. In Proc. of ACM SIGCOMM, pages 89–100, Sept. 1997.

References

[13] C. Huang, R. Janakiraman, and L. Xu. Loss-resilient on-demand media streaming using priority encoding. In Proc. of ACM Multimedia, pages 152–159, Oct. 2004.

[14] L. Juhn and L. Tseng. Harmonic broadcasting for Video-on-Demand service. IEEE Trans. on Broadcasting, 43(3):268–271, Sept. 1997.

[15] H. Ma, G. K. Shin, and W. Wu. Best-effort patching for multicast true VoD service. Multimedia Tools Appl., 26(1):101–122, 2005.

[16] J.-F. Pâris, S. W. Carter, and D. D. E. Long. Efficient broadcasting protocols for video on demand. In Proc. of the Int'l Symp. on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS), pages 127–132, July 1998.

[17] B. Qudah and N. J. Sarhan. Analysis of resource sharing and cache management techniques in scalable video-on-demand. In Proc. of the 14th IEEE Int'l Symp. on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), page 327–334, Sept. 2006.

[18] B. Qudah and N. J. Sarhan. Towards scalable delivery of video streams to heterogeneous receivers. In Proc. of ACM Multimedia, pages 347–356, Oct. 2006.

[19] M. Rocha, M. Maia, I. Cunha, J. Almeida, and S. Campos. Scalable media streaming to interactive users. In Proc. of ACMMultimedia, pages 966–975, Nov. 2005.

References

[20] N. J. Sarhan and C. R. Das. Caching and scheduling in NAD-based multimedia servers. IEEE Trans. on Parallel and Distributed Systems, 15(10):921–933, Oct. 2004.

[21] N. J. Sarhan and C. R. Das. A new class of scheduling policies for providing time of service guarantees in Video-On-Demand servers. In Proc. of the 7th IFIP/IEEE Int'l Conf. on Management of Multimedia Networks and Services, pages 127–139, Oct. 2004.

[22] N. J. Sarhan and B. Qudah. Efficient cost-based scheduling for scalable media streaming. In Proc. of Multimedia Computing and Networking Conf. (MMCN), Jan. 2007.

[23] L. Shi, P. Sessini, A. Mahanti, Z. Li, and D. L. Eager. Scalable streaming for heterogeneous clients. In Proc. of ACM Multimedia, pages 337–346, October 2006.

[24] A. K. Tsiolis and M. K. Vernon. Group-guaranteed channel capacity in multimedia storage servers. In Proc. of ACM SIGMETRICS, pages 285–297, June 1997.

[25] Mohammad Alsmirat, Musab Al-Hadrusi, and Nabil J. Sarhan. Analysis of Waiting-Time Predictability in Scalable Media Streaming. In Proceedings of ACM Multimedia, pages 727 - 736, Augsburg, Germany, September 2007. Acceptance Rate: 19%. DOI: https://doi.org/10.1145/1291233.1291398.

[26] Nabil J. Sarhan, Mohammad A. Alsmirat, and Musab Al-Hadrusi. Waiting-Time Prediction in Scalable On-Demand Video Streaming. ACM Transactions on Multimedia Computing, Communications, and Applications (ACM TOMCCAP), Volume 6, Issue 2, March 2010. DOI: https://doi.org/10.1145/1671962.1671967.